REMARKS

This Amendment is in response to the Final Office Action dated February 14, 2003. In the Office Action, claims 1-10 were rejected under 35 USC §103. By this Amendment, new claims 11-18 are added. Currently pending claims 1-18 are believed allowable, with claims 1, 8, 9, 11, and 15 being independent claims.

CLAIM REJECTIONS UNDER 35 USC §103:

The present invention relates to the scheduling of tasks in data processing systems. Application, page 1, lines 5-7. In some data processing systems, tasks are scheduled by placing pending tasks in a queue. Application, page 1, lines 10-13. When a task is processed, it is taken from the queue and the code for processing the task is loaded into the microprocessor, sometimes via an instruction cache. Application, page 1, lines 13-17. In general, conventional systems process tasks in the natural order in which they become ready for processing. Application, page 7, lines 14-17.

The present invention teaches a technique to optimize the operation of systems that utilize an instruction cache. According to one embodiment of the invention, outstanding tasks are sorted into batches, where each batch consists of tasks requiring the same code to be executed. Application, page 3, lines 1-4. Tasks of like type (i.e., tasks which require the same code path in the instruction cache) are processed in batches such that if a task of a particular type already exists on the queue, then a subsequently scheduled task of the same type is grouped with the existing task instead of being placed at the tail of the queue. Application, page 7, lines 20-26. Such an arrangement beneficially improves system performance because each batch can be processed in one looping operation without incurring instruction cache miss penalties. Application, page 3, lines 7-9.

Claim 1 of the pending application stands rejected as obvious over U.S. Patent No. 5,727,211 to Gulsen ("Gulsen") in light of U.S. Patent No. 5,875,464 to Kirk ("Kirk"). By this Amendment, claim 1 is amended to recite, in part, "placing the tasks of the same task type into a batch such that the tasks in a batch are processed before processing the next ordered task." Support for this amendment can be found at least at page 7, lines 24-26 of the Application. As detailed below, it is respectfully submitted that Gulsen

and Kirk, either along or in combination, do not teach or suggest the above-cited limitation of claim 1.

Gulsen appears to disclose a system and method for context switching between tasks in real time operating systems. Gulsen, col. 1, lines 24-26. The invention compares the current tasks shared system resource utilization requirements to the incoming tasks utilization requirements and swaps the minimum amount of current task information to the systems backing store. Gulsen, col. 1, lines 24-26. Thus, to optimize context switching, the invention swaps out only the portion of the current task's context that would be overwritten by the incoming task. Gulsen, col. 1, lines 4-6. Thus, it is respectfully submitted that Gulsen does not teach or suggest the limitation of placing tasks of the same task type into a batch such that the tasks in a batch are processed before processing the next ordered task, as recited in claim 1 of the present application.

Likewise, Kirk cannot be characterized as teaching or suggesting the above limitation. Kirk appears to disclose a cache system to protect a task's cache region and prevent other tasks in a multitasking environment from perturbing the cache. Kirk, col. 3, lines 34-37. It is respectfully submitted that nowhere in Kirk is there a discussion regarding placing tasks of the same task type into a batch such that the tasks in a batch are processed before processing the next ordered task, as recited in claim 1 of the present application.

In light of the preceding discussion, claim 1 of the present application is not obviated by Gulsen and Kirk, and is believe allowable over the cited art. Furthermore, claims 2-7 are dependent on and further limit claim 1. For at least this reason, claims 2-7 are also believed allowable over the cited art.

Claim 8 of the pending application stands rejected as obvious over Gulsen in light of Kirk. By this Amendment, claim 8 is amended to recite, in part, "code means for scheduling tasks of the same type into a batch such that tasks in a batch are processed before processing the next ordered task." Support for this amendment can be found at least at page 7, lines 24-26 of the Application. As discussed above, it is respectfully submitted that Gulsen and Kirk, either along or in combination, do not teach or suggest the above-cited limitation of claim 8. Thus, claim 8 is believed allowable over the cited art.

Similary, claim 9 of the pending application stands rejected as obvious over Gulsen in light of Kirk. By this Amendment, claim 9 is amended to recite, in part, "means for scheduling tasks of the same type into a batch, wherein the means for processing the tasks is operable to process the tasks in a batch before processing the next ordered task." Support for this amendment can be found at least at page 7, lines 24-26 of the Application. As discussed above, it is respectfully submitted that Gulsen and Kirk, either along or in combination, do not teach or suggest the above-cited limitation of claim 9. Thus, claim 9 is believed allowable over the cited art.

Claims 10 is dependent on and further limits claim 9. For at least this reason, claim 10 is also believed allowable over the cited art.

NEW CLAIMS:

By this amendment, claims 11-18 are added to the present Application. Claim 11 recites, in part, "determining if the task queue includes a cached task that requires substantially the same code to process the cached task and the new task; and batching the new task with the cached task if the task queue includes the cached task that requires substantially the same code to process the cached task and the new task." No new matter is introduced by claim 11 and support for these limitations can be found at least page 3, lines 1-4, page 7, lines 20-26, and Fig. 2 of the present Application. Furthermore, it is respectfully submitted that none of the cited references teach or suggest such limitations. Thus, claim 11 is believe allowable over the prior art.

Claim 12 is dependent on and further limits claim 11. Claim 12 recites, in part, "adding the new task to the end of the queue if the task queue does not include the cached task that requires substantially the same code to process the cached task as the new task." Support for claim 12 can be found at least at page 8, lines 9-10, and Fig. 2 of the present Application. Since claim 11 is believed allowable over the prior art, claim 12 is believed allowable for at least the reasons of claim 11.

Claim 13 is dependent on and further limits claim 11. Claim 13 recites, in part, "loading task code for processing the cached task into an instruction cache; executing the task code for processing the cached task in the instruction cache; and executing the task code for processing the new task in the instruction cache without loading new code into the instruction cache." Support for claim 13 can be found at least at page 8, lines 12-23 of

the present Application. Since claim 11 is believed allowable over the prior art, claim 13 is believed allowable for at least the reasons of claim 11.

Claim 14 is dependent on and further limits claim 13. Claim 14 recites, in part, "determining if the task code is capable of fully loading into the instruction cache; and if the task code is not capable of fully loading into the instruction cache, logically dividing the task code such that at least one substantially atomic portion of the task code will fully load in the instruction cache." Support for claim 14 can be found at least at page 3, lines 14-30 of the present Application. Since claim 13 is believed allowable over the prior art, claim 14 is believed allowable for at least the reasons of claim 14.

Claim 15 recites, in part, "computer readable program codes configured to cause the program to: identify a new task to be scheduled in the task queue; determine if the task queue includes a cached task that requires substantially the same code to process the cached task and the new task; and batch the new task with the cached task if the task queue includes the cached task that requires substantially the same code to process the cached task and the new task." Support for these limitations can be found at least page 3, lines 1-4, page 7, lines 20-26, and Fig. 2 of the present Application. Furthermore, it is respectfully submitted that none of the cited references teach or suggest such limitations. Thus, claim 15 is believe allowable over the prior art.

Claim 16 is dependent on and further limits claim 15. Claim 16 recites, in part, program code "configured to cause the program to add the new task to the end of the queue if the task queue does not include the cached task that requires substantially the same code to process the cached task as the new task." Support for claim 16 can be found at least at page 8, lines 9-10, and Fig. 2 of the present Application. Since claim 15 is believed allowable over the prior art, claim 16 is believed allowable for at least the reasons of claim 15.

Claim 17 is dependent on and further limits claim 15. Claim 17 recites, in part, program code "configured to cause the program to: load task code for processing the cached task into an instruction cache; execute the task code for processing the cached task in the instruction cache; and execute the task code for processing the new task in the instruction cache without loading new code into the instruction cache." Support for claim 17 can be found at least at page 8, lines 12-23 of the present Application.

Since claim 15 is believed allowable over the prior art, claim 17 is believed allowable for at least the reasons of claim 15.

Claim 18 is dependent on and further limits claim 17. Claim 18 recites, in part, program code "configured to cause the program to: determine if the task code is capable of fully loading into the instruction cache; if the task code is not capable of fully loading into the instruction cache, logically divide the task code such that at least one substantially atomic portion of the task code will fully load in the instruction cache." Support for claim 18 can be found at least at page 3, lines 14-30 of the present Application. Since claim 17 is believed allowable over the prior art, claim 18 is believed allowable for at least the reasons of claim 17.

CONCLUSION

In view of the forgoing remarks, it is respectfully submitted that this case is now in condition for allowance and such action is respectfully requested. If any points remain at issue which the Examiner feels could best be resolved by a telephone interview, the Examiner is urged to contact the attorney below.

This Amendment is being filed with a petition for a three month extension of time under 37 CFR 1.136(a). A check including the $930 for the three-month extension of time fee, $750 the RCE fee, and $168 the two additional independent claims fee is also included. No additional fee is believed due with this Amendment, however, should a fee be required please charge Deposit Account 50-0510.

Respectfully submitted,

Dated: August 15, 2003

Ido Tuchman, Reg. No. 45,924
Law Office of Ido Tuchman
69-60 108th Street, Suite 503
Forest Hills, NY 11375
Telephone (718) 544-1110
Facsimile (718) 544-8588